**1. Write a C program for Caesar cipher involves replacing each letter of the alphabet with**

**the letter standing k places further down the alphabet, for k in the range 1 through 25.**

#include <stdio.h>

int main() {

char plaintext[100];

int k, i;

printf("Enter a message to encrypt: ");

fgets(plaintext, sizeof(plaintext), stdin);

printf("Enter shift value (1-25): ");

scanf("%d", &k);

if (k < 1 || k > 25) {

printf("Shift value must be between 1 and 25.\n");

return 1;

}

printf("Encrypted message: ");

for (i = 0; plaintext[i] != '\0'; i++) {

char ch = plaintext[i];

// Encrypt uppercase letters

if (ch >= 'A' && ch <= 'Z') {

ch = ((ch - 'A' + k) % 26) + 'A';

}

// Encrypt lowercase letters

else if (ch >= 'a' && ch <= 'z') {

ch = ((ch - 'a' + k) % 26) + 'a';

}

Putchar(ch);

}

printf("\n");

return 0;

}

**2.Write a C program for monoalphabetic substitution cipher maps a plaintext alphabet to a**

**ciphertext alphabet, so that each letter of the plaintext alphabet maps to a single unique**

**letter of the ciphertext alphabet.**

#include <stdio.h>

#include <string.h>

int main() {

char plaintext[100], ciphertext[100];

char \*key = "QWERTYUIOPASDFGHJKLZXCVBNM";

char \*alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

printf("Enter plaintext (uppercase letters only): ");

fgets(plaintext, sizeof(plaintext), stdin);

plaintext[strcspn(plaintext, "\n")] = 0;

for (int i = 0; plaintext[i] != '\0'; i++) {

if (plaintext[i] >= 'A' && plaintext[i] <= 'Z') {

int index = plaintext[i] - 'A';

ciphertext[i] = key[index];

} else {

ciphertext[i] = plaintext[i];

}

}

ciphertext[strlen(plaintext)] = '\0';

printf("Ciphertext: %s\n", ciphertext);

return 0;

};

**3. Write a C program for Playfair algorithm is based on the use of a 5 X 5 matrix of letters**

**constructed using a keyword. Plaintext is encrypted two letters at a time using this matrix.**

#include <stdio.h>

#include <string.h>

void createMatrix(char key[], char matrix[5][5]) {

int k = 0, i, j;

int alpha[26] = {0};

for (i = 0; key[i] != '\0'; i++) {

if (key[i] >= 'A' && key[i] <= 'Z' && alpha[key[i] - 'A'] == 0) {

matrix[k / 5][k % 5] = key[i];

alpha[key[i] - 'A'] = 1;

k++;

}

}

for (i = 0; i < 26; i++) {

if (alpha[i] == 0 && (i + 'A' != 'J')) {

matrix[k / 5][k % 5] = i + 'A';

k++;

}

}

}

void encrypt(char plaintext[], char matrix[5][5], char ciphertext[]) {

int i, j, k = 0;

for (i = 0; plaintext[i] != '\0'; i += 2) {

if (plaintext[i + 1] == '\0') {

plaintext[i + 1] = 'X'; // Padding

}

int row1, col1, row2, col2;

for (row1 = 0; row1 < 5; row1++) {

for (col1 = 0; col1 < 5; col1++) {

if (matrix[row1][col1] == plaintext[i]) {

break;

}

}

if (col1 < 5) break;

}

for (row2 = 0; row2 < 5; row2++) {

for (col2 = 0; col2 < 5; col2++) {

if (matrix[row2][col2] == plaintext[i + 1]) {

break;

}

}

if (col2 < 5) break;

}

if (row1 == row2) {

ciphertext[k++] = matrix[row1][(col1 + 1) % 5];

ciphertext[k++] = matrix[row2][(col2 + 1) % 5];

} else if (col1 == col2) {

ciphertext[k++] = matrix[(row1 + 1) % 5][col1];

ciphertext[k++] = matrix[(row2 + 1) % 5][col2];

} else {

ciphertext[k++] = matrix[row1][col2];

ciphertext[k++] = matrix[row2][col1];

}

}

ciphertext[k] = '\0';

}

int main() {

char key[100], plaintext[100], ciphertext[100];

char matrix[5][5];

printf("Enter the keyword: ");

gets(key);

printf("Enter the plaintext: ");

gets(plaintext);

createMatrix(key, matrix);

encrypt(plaintext, matrix, ciphertext);

printf("Ciphertext: %s\n", ciphertext);

return 0;

}